



RECEIVED

JAN 22 2002

TECHNOLOGY CENTER 2800

*Exhibit A*



**Synplicity Floorplanner Product Overview**  
(HDL Floorplanner)  
January, 1998

**COPY**

RECEIVED  
JAN 22 2002  
TECHNOLOGY CENTER 2800

#### Overview

The timing delay incurred by IC wiring interconnect now far exceeds the delay of the logic itself. This fact has broken traditional design flows consisting of HDL -> Synthesis -> Place & Route. Designers of the largest ASICs have been helped out by the introduction of floorplanning tools which help guide the place & route process and give better predictability to the timing of the ASIC. The latest FPGA devices have a similar problem making it necessary to consider the physical placement of logic in the finished device. This consideration should occur during the synthesis process and before the placement and routing of the device. An in-depth knowledge of the placement tools used in the back-end is necessary to accurately estimate wire lengths which can be translated to timing delay and presented to a designer before going through time-consuming place & route.

Additionally, the threshold (in terms of gate count) for floorplanning FPGAs is lower than with ASICs since the ratio of routing resources to logic resources is lower for FPGAs than ASIC. For these reasons, FPGA floorplanning is needed now for even some of the smaller devices that are more structured (datapath-intensive) and will become essential for the largest devices available within 1 to 2 years.

#### Key Problems to be solved

- 1) Managing & guiding the physical implementation of large FPGAs with lots of hierarchy
  - Understand the different FPGA architectures / logic structure for floorplanning
  - Create physical hierarchy based upon initial RTL and additional user guidance
- 2) Minimize iterations of P&R for high-density FPGAs
  - Accurate estimation of timing before performing P&R
  - Pass users timing and placement constraints to FPGA Vendor's P&R tools
    - Driving delay prediction
  - Report on chip performance (critical paths).
    - Answering; will the device make speed?
    - Know which blocks are fast and which are slow
- 3) Answering; Can the design be routed in a specific device?
  - Reports on routability (congestion analysis)
  - If not, show what device can the design ~~can~~ be routed in
- 4) Shorten place & route time
  - Average P&R for high-density designs is ~3hrs Xilinx & ~1.5hrs Altera
- 5) Provide an incremental methodology for floorplanning where blocks that are well characterized may be "frozen" and not changed by subsequent floorplanning

#### Product Objectives

- 1) Position Synplicity as *the* leader for next-generation FPGA design methodology by having the first general-purpose (non-vendor specific) FPGA floorplanner
- 2) Initially focus on Altera and Xilinx devices (both available upon initial release, with support for other vendors added over time as needed.
- 3) Leverage work between Multi-chip partitioner and FPGA floorplanner
- 4) First release available by June '98 (beta testing begins in April '98)
- 5) Provide a new product for sales and increase the average revenue generated per seat by 2X

#### Target Devices

By far the most requested vendors where floorplanning is considered to become necessary are Altera and Xilinx. Of the high-density FPGA designers surveyed, roughly 35% believe they need floorplanning now with another 50% saying they will need it within one year. Datapath-intensive applications using relatively small devices (20K gate range) can benefit significantly in terms of performance when they are floorplanned. Specifically the devices requested were:

Altera Flex 10K50's and higher  
Xilinx XC4044XLs and higher  
Xilinx XC40125XV's and higher

Altera's Raphael and Xilinx' Virtex will also require floorplanning and should be supported by the floorplanner as soon as they become available from the vendors. The Lucent Orca/3C devices are also high capacity and will most likely be the next vendor to be supported after Altera and Xilinx.

#### Product Description (High-level)

The floorplanner will be a graphical tool that takes the RTL (Verilog or VHDL) compiled by Synplify and then facilitates floorplanning by allowing the user to assign HDL modules to specific areas on the target device. As blocks of HDL are moved into the physical space, the Synplify synthesis engine will perform accurate estimation of block size and performance. Synthesis and floorplanning may be performed in parallel. The user can start top-down with empty HDL blocks and fill in the HDL incrementally or bottom up by floorplanning complete HDL blocks from the start. Routability and performance information will then be presented to the user in various forms.

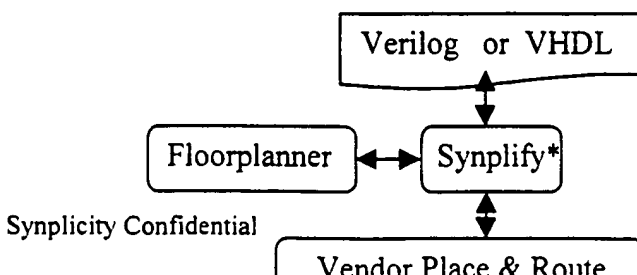
#### GUI Summary

- 1) Graphically display HDL hierarchy via a Windows browser type method
  - If a block has not yet been developed, the user can specify an estimated size and add the HDL code to it later.
- 2) Provide drag (HDL module from browser) and drop into physical hierarchy for editing
- 3) Allow the user to provide guidance for:
  - pins (pin-locking)
  - multiple clocks
  - clock skew
  - I/Os
  - timing-driven placement (path constraints)
  - bus structure (i.e. interleaving) – *Later release*
  - CLB placement (for Xilinx) – *Later release*
  - "Soft grouping" of related logic – With automatic floorplanner in a *later release*
- 4) Call synthesis and provide user feedback:
  - display amount of capacity left in the partitions
  - display timing estimates for critical paths in the module
- 5) Re-entrant/incremental floorplanning is required. This allows specified blocks to be frozen (not synthesized or changed in any way) once their performance and size goals have been met, but other blocks within the chip still need work. This avoids the problem of having the tools change "known good" blocks when a change is made in another part of the chip.
- 6) Pass floorplanning information to the target place & route environment.

#### User Feedback Prioritization:

Below are the responses from the customers surveyed regarding their prioritization of the type of feedback the floorplanner tool should focus on. Users had 100 points to assign to each of the 5 categories below. Routability and timing are top priorities. Need to provide at minimum feedback on these two items.

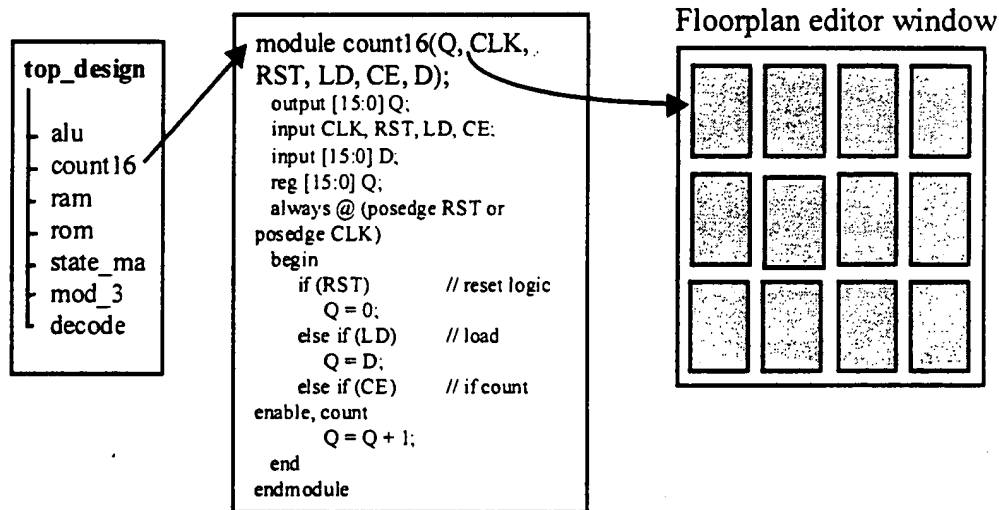
<u>Feedback on</u>	<u>Ave.Priority</u>	<u>Range of responses</u>
Routability	34.4	(9 to 75)
Timing	29.6	(15 to 40)
Utilization	14.7	(1 to 33)
Power	9.4	(0 to 30)
Clock Skew	9.6	(0 to 30)



### Feature Requests:

Below are a few of the specific feature requests regarding floorplanning that came up frequently during the interviews that should be considered with the initial product.

- 1) Set and display clock frequency and other constraints on a block by block basis.  
Perform optimization on a block by block basis (i.e. the user can say one block is timing critical so optimize it for timing, while another is not, so optimize it for area)
- 2) Show slack time for anything that the user puts constraints on.
- 3) Automatically pass floorplanning constraints to Place & Route tools
- 4) Provide routability analysis based upon a user-specified utilization factor. For example, if the user knows that 75% utilization for a certain part is good, he'd like to be able to specify this factor (75%) and have the tool provide feedback based upon this factor.
- 5) Provide mechanism to allow bus interleaving and register grouping – *Later release*
- 6) Allow I/O and general pin alignment
- 7) Provide "locate" command to find specific nets
- 8) Display "Ratsnest" graphical display for connectivity analysis
- 9) Tell the user if the design is routable in a particular part
- 10) Floorplanner "must" use same placement algorithm as target P&R tool – Cisco (ASIC)  
- must have P&R timing accuracy within 20% - 10% to 15% would be better
- 11) Extract parasitic information and put into PDEF – *Later release*
- 12) Ability to place the same objects in multiple locations (replication of logic)



### Competition

There is not much direct competition for FPGA floorplanning at this time although we must assume that Exemplar and Synopsys are aware of this need and will be addressing it in 1998. The current floorplanners for programmable devices available today are from Xilinx and Morphologic.

**Xilinx – XACTstep** floorplanner includes capabilities for:

- 1) Drag & Drop elements into the floorplan
- 2) Hand place critical elements
- 3) Analyze net congestion
- 4) Rip up and redo routing

See: [http://www.xilinx.com/products/software/xact\\_fp.pdf](http://www.xilinx.com/products/software/xact_fp.pdf) for details

**Morphologic – MorphMCFP** provides the ability to:

- 1) Floor plan multiple devices simultaneously
- 2) Floor plan devices of different technologies simultaneously
- 3) Floor plan all Xilinx XC4000 and 6200 parts
- 4) Floor plan all Lucent 2C and 2T devices
- 5) Use physical interconnect from existing FPGA-based circuit cards to support partitioning across multiple devices.

See <http://www.morphologic.com/mmcfp.htm> for details

### **Target Market**

The primary market for the floorplanner will be for designers of high-density FPGAs. The initial release of the floorplanner will target Xilinx and Altera high-end users. Support for other vendors will follow as the market demands. Primary device support includes:

Altera Flex10K50s and above + Raphael Devices (when available from Altera)  
Xilinx XC4044XLs and above + XC4000XVs + Virtex

Xilinx estimates that their High Density business alone (XL, XV, Virtex plus futures) will be ~\$100 million by the year 2000

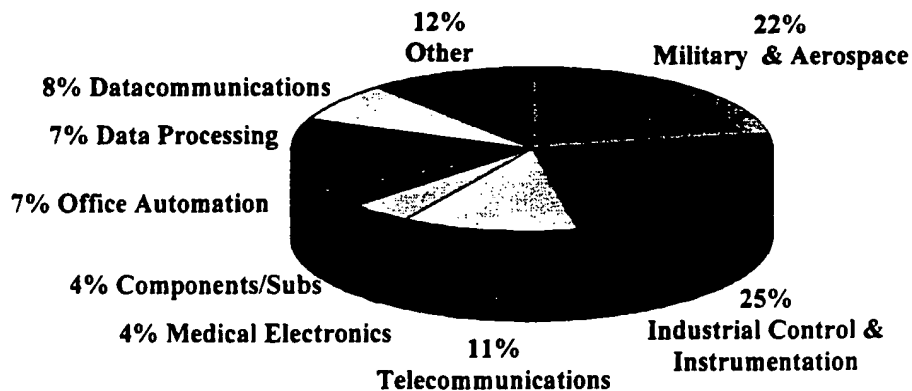
Applications segments that tend to utilize the largest programmable devices are:

*Communications*; Cisco, Bay Networks and other companies with networking and communications core competence are extremely sensitive to design cycle time and as such require large quantities of high-end FPGAs to bring products to market as early as possible. Deep submicron effects in the large FPGAs will prolong these cycles without tools to do timing-driven design and converge on a timing solution quickly.

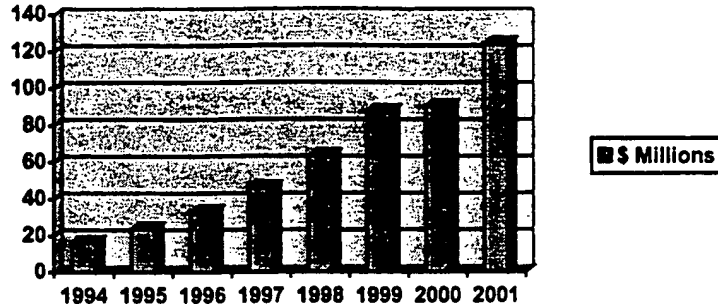
*Graphics processing & Imaging*; Graphics processing and application specific DSP designers are increasingly dependent upon FPGAs to implement signal processing algorithms in hardware. Mil/Aerospace and real-time control applications commonly use dedicated hardware for data collection and display.

Below is a breakdown of FPGA designs by application segments from the 1996 EE Times PLD research study. The breakdown remains essentially the same for 1997 with the communications segment gaining the most. The communications segments are also where the large designs tend to reside. Industrial control and military applications go more for the smaller devices.

**PLD Users Applications**



Floorplanner Revenue History and Forecast (ASIC) has 30.4% CAGR. We can assume a similar growth rate for FPGA floorplanners once they become available.



#### Documentation

As this will be the first EDA-vendor provided floorplanning tool for FPGAs and the methodology for such a tool is not known by its intended users, it will be necessary to provide a comprehensive users guide & tutorial describing the design flow and use model. The user guide should outline an approach / methodology for integrating the floorplanner into existing tool flows and guidelines for using the floorplanner effectively.

#### Platform Support

The floorplanner will need to be supported on the standard Synplicity hardware platforms. Special attention should be paid to the GUI on UNIX machines as the companies that will be using floorplanning heavily will tend to be the larger accounts where UNIX machines are still dominant. As with all Synplicity software after and including the 3.1 release, Windows 3.1 will not be supported.

PC – Windows 95/98 and Window NT  
 Sun – Solaris  
 Sun – SunOS  
 HP – HP-UX

#### Batch Operation

The floorplanner is meant to be an interactive tool that requires user input to select HDL modules for floorplanning. The commands that are available through the GUI should also however be available via tcl script so users may include floorplanning commands in their scripts that take a design through the entire flow (floorplanning, synthesis, place & route, etc.). Again, this is important for the larger UNIX shops that use batch operation as their primary mode of operation.

#### Packaging & Pricing

The floorplanner will be a complimentary system for performing floorplanning for FPGA devices. It will have its own product number, but will require the user to also purchase Synplify. It will not be supported with synthesis tools other than Synplify. As such, a separate license key will be required to run the floorplanner.

Product Number	Description	US Price
SFPN*	Synplicity Floorplanner for PC Node-locked	\$17,000
MFPN	Maintenance for Floorplanner Node-locked	\$3,400
SFAF**	Synplicity Floorplanner Floating license	\$34,000
MFAF	Maintenance for Floorplanner Floating	\$6,800

\*requires SSPN

\*\*requires SSAF

**Target Schedule**

Product release	June 1998
Beta testing begins	April 1998
Prototype demonstration	?